K-Nearest Neighbors

The Power of Simplicity in Machine Learning

Ahmed BADI

Mathematics & Machine Learning Enthusiast

ahmedbadi905@gmail.com
linkedin.com/in/badi-ahmed

December 15, 2025

Overview

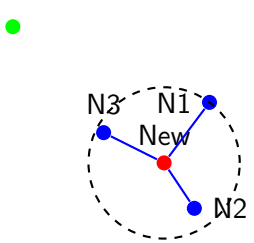
Introduction

The Intuition: "Birds of a Feather"

- **Scenario:** You move to a new neighborhood. How do you guess the local favorite sports team?
- **Strategy:** Look at your 3 closest neighbors.
- **Core Principle:** Similar things exist in close proximity.

What is KNN?

A supervised learning algorithm that makes no assumptions about data distribution (Non-Parametric) and memorizes the training set (Lazy Learner).



figureKNN classification: The red point is classified based on the 3 nearest neighbors (blue).

The Core Algorithm

How KNN Works (4 Steps)

Given a new data point x_{new}:

1. **Choose K:** Pick the number of neighbors (e.g., $K = 5$).
2. **Calculate Distances:** Measure distance from x_{new} to all training points.
3. **Find Neighbors:** Sort and pick the top K closest points.
4. **Vote or Average:**
   - *Classification:* Majority Vote.
   - *Regression:* Average value.

Visualizing the Decision

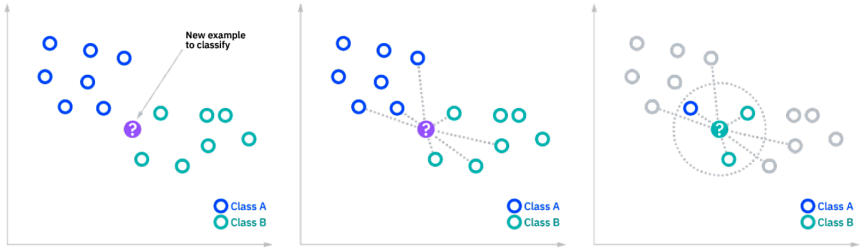


Figure: Impact of K: With K=3, Choice of K changes the outcome.

Mathematics of Closeness

Distance Metrics

How do we measure "closeness"?

Euclidean Distance (Straight Line)

Default for continuous data.

$$d(p, q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$$

Manhattan Distance (Taxicab)

Better for high dimensions (Grid movement).

$$d(p, q) = \sum_{i=1}^{n}|q_i - p_i|$$

Other Metrics

- **Minkowski Distance:** Generalization of Euclidean ($p = 2$) and Manhattan ($p = 1$).

$$d(p, q) = \left(\sum |q_i - p_i|^p \right)^{1/p}$$

- **Hamming Distance:** For categorical variables (strings, bits). Counts mismatches.

Choosing K & Preprocessing

The Bias-Variance Tradeoff

Small K (e.g., K=1)

- High Variance (Overfitting).
- Sensitive to noise/outliers.
- Complex decision boundaries.

Large K (e.g., K=100)

- High Bias (Underfitting).
- Overly smooth boundaries.
- Ignores local patterns.

Solution: Use Cross-Validation (Elbow Method) to find the sweet spot.

The Bias-Variance Graph

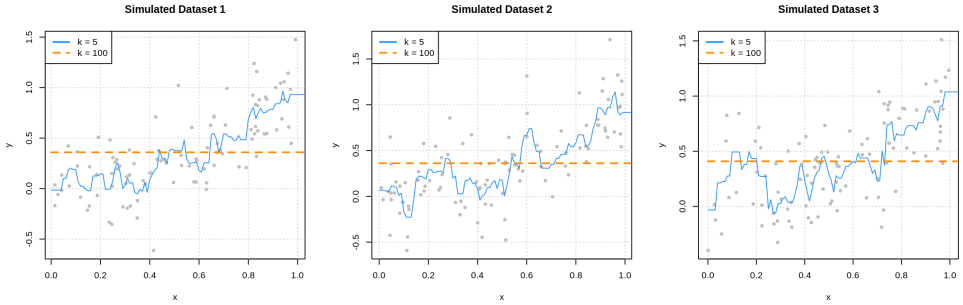


Figure: Bias Variance Tradeoff.

Feature Scaling is Critical!

Warning

KNN depends on distance. Features with large numbers dominate features with small numbers.

Example:

- Salary: 20,000 to 200,000 (Range = 180,000)
- Age: 20 to 60 (Range = 40)

Euclidean distance will effectively ignore Age.

Solution: Min-Max Normalization or Z-Score Standardization.

Optimization Challenges

Beyond Brute Force

Standard KNN calculates distance to *every* point: $O(N \cdot D)$. Slow for big data.

KD-Trees (k-Dimensional Trees)

- Partition space into hyper-rectangles.
- Allows skipping entire branches of distant points.
- Complexity drops to $O(\log N)$.

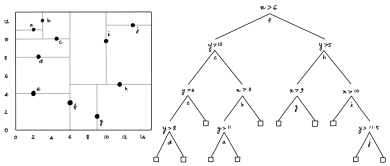


Figure: KD-Tree Splitting.

The Curse of Dimensionality

The Problem

As dimensions increase ($D \to \infty$):

1. Data becomes incredibly sparse (empty space).
2. All points become equidistant. "Nearest" loses meaning.

Mitigation: Dimensionality Reduction (PCA, t-SNE) before KNN.

Conclusion

Summary: Pros and Cons

Advantages

- Intuitive and simple.
- No training phase (Lazy).
- Adapts to new data instantly.
- Inherently multi-class.

Disadvantages

- Slow prediction phase.
- Memory intensive.
- Sensitive to scaling & noise.
- Fails in high dimensions.

Thank You!

"Simplicity is the ultimate sophistication."

Questions?